Journal of Intelligent Information Systems, 11, 259–285 (1998)

Knowledge Mining With VxInsight: Discovery Through Interaction

GEORGE S. DAVIDSON gsdavid@sandia.gov
BRUCE HENDRICKSON bahendr@sandia.gov
DAVID K. JOHNSON dkjohns@sandia.gov
CHARLES E. MEYERS cemeyer@sandia.gov
BRIAN N. WYLIE bnwylie@sandia.gov
Sandia National Laboratories, Albuquerque, NM

Abstract. The explosive growth in the availability of information is overwhelming traditional information management systems. Although individual pieces of information have become easy to find, the larger context in which they exist has become harder to track. These contextual questions are ideally suited to visualization since the human visual system is remarkably adept at interpreting large quantities of information, and at detecting patterns and anomalies. The challenge is to present the information in a manner that maximally leverages our visual skills. This paper discusses a set of properties that such a presentation should have, and describes the design and functionality of VxInsight, a visualization tool built to these principles.

Keywords: information visualization, information retrieval, graphical user interface, browsing

1. Introduction

For most of history, mankind has suffered from a shortage of information. Now, in just the infancy of the electronic age, we have begun to suffer from information excess. Data overload is bound to get worse. An urgent need exists for tools to help manage, extract information and accumulate knowledge from increasingly large collections of data, now being constructed and made available from data warehouses. Existing tools to address these problems are often hard to use, incomplete and generally inadequate.

As we have begun to use computers to manage information, it probably can't be helped that our first thoughts are to automate the way we have done things in the past. This is the basis for online card catalogs, computerized abstract and title index services, and other tools that greatly speed up the steps we have traditionally used to work with libraries and other large databases. However, as Bar and Borrus have noted, "one reason Information Technology investments have not translated into higher productivity is that they have primarily served to automate existing tasks. They often automate inefficient ways of doing things" [3]. While such approaches are of undeniable value, they have proved insufficient to solve the problem of information overload. New insights and techniques are required. Fortunately, they can be enabled by the same technological advances that created the glut of data.

One of the key shortcomings of information management systems is in the nature of the user interface. Interpreting textual information is a sequential and slow process, which imposes a limit on the speed with which information can be assimilated and processed. Text is excellent for conveying detailed information, but it is poorly suited for conveying

relationships or trends or for getting an overview of a set of data. For these tasks, graphical displays are more effective and have become essential elements of scientific research [25].

Graphical displays can be effective at conveying information because of the remarkable capabilities of the human visual cortex. Compared to the eons of evolutionary progress in vision, our talents for interpreting text and speech are recent and primitive. The visual system is incredibly effective at identifying trends and patterns, and detecting anomalies in large sets of data. These innate skills have the potential to significantly alleviate the problem of information overload. For this potential to become a reality, graphical interfaces must be developed for all types of abstract information. These interfaces must be carefully designed to leverage the inherent talents of our visual systems. However, they must also be modest, only exploiting visualization where appropriate, because many kinds of information are best conveyed through other channels. We have attempted to build such a tool, initially to address specific questions, but which we have come to realize is very broadly applicable.

Our VxInsight tool is a graphical interface to large databases. It presents a visual representation of the data elements in which the geometric placement of the objects conveys significant information. Related database objects are located on a 2-D plane with a proximity based on a measure of their similarities. The resulting aggregation of elements into groups is the *context* within which an analyst begins to understand *the implicit structure of the data, rather than just asking about its content.* Context is a critical component of knowledge, which is easily lost in a deluge of raw data. Fortunately, we can represent detail while simultaneously displaying context in a graphical manner.

VxInsight overlays the 2-D plane with a 3-D virtual landscape that looks like a mountain range. The height of a mountain is proportional to the density of objects beneath it. This 3-D environment is readily explored because there is only a small cognitive step between seeing the presented images and then exercising our innate human expertise in navigating through real landscapes.

VxInsight allows one to see the big picture, to zoom in and see the details right down to the network of relationships among the data elements. If the database represents data that has accumulated through time, VxInsight also allows the analyst to see the terrain representations evolve temporally as the data set increases through its collection lifetime. In our experience, these views of the data can be highly informative. But the real power of VxInsight becomes apparent when the analyst begins to ask questions using the Standard Query Language (SQL) interface to the database. Typically, the analyst will compose a Boolean query that is passed back to the database engine. VxInsight uses the response from the database engine to visually mark the data elements that match the query. It does so by coloring spots (on the 3-D terrain representation) above the matched data elements. The results of each query are shown with different colors. This places query results in the context of the overall information terrain, and in relation to other queries, thereby providing a powerful way to answer questions. This representation shows not only the matching contents of the database, but also says important things about the intersection between the analyst's question and the implicit structure of the database. Hypotheses can be formulated, tested and revised, and the answers can suggest new questions – all in real time. It is this participatory interaction with the data that gives VxInsight its power.

In the next section we present a more detailed overview of VxInsight and discuss its relationship to previous work. In Section 4 we discuss placing the data objects into the 2-D

plane and present several algorithms for this problem. In Section 5 we describe how an analyst uses the system. We follow with a discussion of current and future applications in Section 6, and conclusions in Section 8.

2. An Overview of VxInsight

VxInsight provides a visual mechanism for browsing, exploring and retrieving information from a database. The graphical display conveys information about the relationship between objects in several ways and on multiple scales. In this way, individual objects are always observed within a larger context.

For example, consider a database consisting of a set of scientific papers. Imagine that the papers have been organized in a two dimensional geometry so that related papers are located close to each other. Now construct a landscape where the altitude reflects the local density of papers. Papers on physics will form a mountain range, and a different range will stand over the biological papers. In between will be research reports from biophysics and other bridging disciplines. Now, imagine exploring these mountains. If we zoom in closer, the physics mountains will resolve into a set of sub-disciplines. Eventually, by zooming in far enough, the individual papers become visible. By pointing and clicking you can learn more about papers of interest or retrieve their full text.

Although physical proximity conveys a great deal of information about the relationship between documents, you can also see which papers reference which others, by drawing lines between the citing and cited papers. For even more information, you can choose to highlight papers by a particular researcher or a particular institution, or show the accumulation of papers through time, watching some disciplines explode and other stagnate.

VxInsight is a general purpose tool, which enables this kind of interaction with wide variety of relational data: documents, patents, web pages, and financial transactions are just a few examples. The tool allows users to interactively browse, explore and retrieve information from the database in an intuitive way.

Several basic principles guided the design of the tool.

1. The human visual system excels at identifying patterns, trends and anomalies.

2. A human can better interpret data with presentations built on familiar metaphors.

3. A useful tool allows easy and intuitive navigation which provides both detailed and high-level views.

4. Comprehension of large datasets is facilitated by the ability to explore the data at different resolutions, for instance, by zooming into and out of the representation. To enhance interpretability, the resolution should vary continuously.

5. Since no single representation is optimal for all types of questions, a tool should contain a tool kit of different display and interaction mechanisms.

6. Graphical displays should have compact representations to facilitate interactive frame rates and access over a network. However, simplicity must not prevent the tool from being applicable to large datasets.

In VxInsight, we use geometric proximity as a metaphor showing the relatedness of two objects, and present the result as 3-D landscape. This is a very intuitive mechanism, but it has significant limitations. Databases typically encode many different kinds of relationships, so proximity in a low-dimensional space is insufficient to fully capture all of the complexity in the relationships. Although placing the objects in a higher-dimensional geometric space would allow proximity to convey more information, users have great difficulty interpreting and navigating higher dimensional spaces.

We feel the ease of interaction in 3-D compensates for this compression of information. Nevertheless, the process used to determine where objects will be placed is critical to the quality of information preserved for presentation. We discuss this process in Section 4.

2.1. Related Work

A number of research efforts and commercial applications have applied visualization techniques to databases. Many of these efforts have been domain-specific, but some have been more general purpose. We will not survey this rapidly expanding field, but we will review the projects that overlap with VxInsight in philosophy or in approach.

For commercial products that strive to extract useful information from large collections of data, see, for instance the comparison articles [8] and [10]. None of these commercial products make intensive use of high-performance graphical displays. Most of the interesting research in visualizing abstract information hasn't yet penetrated the commercial market, although a number of recent startups may change this in the near future.

In [20], four types of visual displays for information retrieval are described. These are hierarchical, network, scatter and map displays. Three of these types are contained within VxInsight. As will be detailed in Section 5, links between objects can be displayed as appropriate, which constitute a network display. The zoom capability within VxInsight produces a dynamic hierarchical view of the data. The third and principle view within VxInsight is closely related to scatter displays. In traditional scatter displays, data objects are represented as points in a two-dimensional plane. But this representation becomes intolerably crowded when there are a large number of objects. Our solution to this problem is to display the density of objects as a landscape, only displaying individual objects when requested or when the user has zoomed in so that there are only a modest number of objects to display. A number of other projects have addressed this problem in different ways.

For example, [6] suggests mapping objects to points in 3-D, but only visualizing 2-D projections. The analyst can then control the projection to identify the features of greatest interest, or to find the most informative display. The flexibility of this approach is attractive, and more information can be encoded in the proximity relationships in 3-D. But if the dataset is large, the fundamental problem of an overly crowded display remains.

The Visual Insights project from Lucent Technologies [26] has more in common with our approach. They use a 2-D scatter or network display, but they enable the analyst to zoom into regions of interest. This substantially resolves the overcrowding issue. However, without any mechanism for identifying the different screen regions, navigation is difficult; an analyst has trouble knowing where to zoom. But, as we will discuss in Section 5, dynamic peak labels in our landscape provide important navigational guidance.

Several tools have been developed that extend scatter displays to three dimensions by placing a spike on top of each object. Two examples are the MineSet software from SGI [22] and the SDM package from Carnegie Mellon [5]. The height of the spike can be used to convey information about the objects, and navigation in 3-D enables exploration of the data. As with the Visual Insights tool, no capacity for automatically identifying areas of the display are included.

Other projects, developed independently and concurrently with VxInsight, have taken the approach we advocate: forming a density-based landscape. In the work of [11], a landscape representing the World Wide Web is constructed in much the same way as we would do with VxInsight. First, the objects are positioned in the plane using a self-organizing map approach. In contrast to our approach, a self-organizing map restricts object locations to lattice points. Next, density is used to specify an altitude for that region of the landscape. However, zooming and peak labeling techniques are absent in this work. A closely related effort is the WEBSOM project [16, 15]. Self-organizing maps are again used to position objects, but instead of a 3-D landscape, a 2-D display is produced in which color represents density. However, peak labels are automatically generated, and some very limited navigational and retrieval capabilities are provided.

The work most similar to our own is the SPIRE project [28], which originated at the Pacific Northwest National Laboratory and is now being commercialized by ThemeMedia. Like VxInsight, SPIRE maps objects to a two-dimensional plane so that related objects are near each other and then provides graphical interfaces for interaction. SPIRE is exclusively focused on textual objects and computes similarities using text analysis. For ordination, SPIRE uses several clustering approaches, or, for small data sets, multidimensional scaling. SPIRE has two visualization approaches. The first, and primary view is a scatter plot of the text documents. The second view is a high level terrain display similar to our own. However, unlike VxInsight, the resolution of the terrain view is completely static, there is no ability to zoom into the mountains to see more structure. SPIRE does not provide the continuous, multi-resolution viewing capability that we feel is essential for revealing the inherent hierarchical structure of the data. In addition, SPIRE lacks the flexible database interface which is a central component of VxInsight.

In summary, we are not aware of any other tool for visualizing databases which has the interactivity and flexibility of VxInsight. Although we fell this adds significant power to the tool, it also leads to new challenges and algorithmic questions. Since VxInsight supports continuous resolution of the landscape, most of the calculations and visualization must be done in real time as a user navigates through the data. The landscape is constructed, peak labels are generated and the terrain is visualized on the fly. None of this can be precomputed. Thus, we need very fast algorithms for each of these steps. Fortunately, the degree of interactivity reduces the quality demanded of individual images. So the algorithmic challenge is to find the best possible solution within stringent time constraints. Our solutions to these problems are discussed in Section 5. We don't claim to have the final word on any of these issues and hope that VxInsight will help motivate future work leading to improved approaches.

3. Geometric Placement of the Objects

A guiding motivation behind VxInsight is the use of geometric proximity in the graphical user interface to capture relationships between objects. Thus, a critical step is the mapping of objects to geometric coordinates in such a way that related documents are kept close together. The ability to extract useful information from the tool depends upon the quality of this *ordination*. The properties of a good ordination depend upon the nature of the data and the questions being asked, so it is our conviction that no single approach is a panacea. With domain-specific expertise, an analyst may be able to provide a better ordination than any general purpose tool. For this reason, we allow the analyst to choose among several ordination algorithms, or to bypass our algorithms and provide coordinates directly.

3.1. The Similarity Function

Input to the ordination algorithms in VxInsight consists of a *similarity* function $s : O \times O \rightarrow \mathbb{R}$, which maps object pairs to non-negative real numbers. Larger values imply that the two objects are more similar and hence should be located closer together. For large datasets, we expect most of the similarity values to be zero, so the data representing the similarity relationship will be sparse.

The details of the similarity function depend upon the application. Some simple examples of the data that could be used for similarity generation include the following.

1. Common keywords in documents.
2. Identical vocabulary within documents.
3. Citation links between scientific papers or patents.
4. Direct links in web documents.
5. Financial transaction links between corporations.
6. Membership in common organizations among individuals.

More generally, because any relational database includes information that couples objects, the different relational fields can be summed or combined in more complex ways to generate similarity values. Of course, more sophisticated methodologies, like latent semantic indexing, [7] could also be used. Obviously, the analyst should select a function that is appropriate for the kinds of questions being addressed.

3.2. Classes of Ordination Algorithms

Given a similarity function, the goal of the ordination process is to place the objects in a geometric space so that similar items are close together, and dissimilar objects are far apart. As we discussed above, for VxInsight we want a two-dimensional ordination, but the problem can be phrased in any dimension. Various approaches have been proposed for this and closely related problems; see, for example, the literature on self-organizing maps, eg.

[18]. Much of the prior work can be grouped into one of the three categories we discuss below. However, none of these general approaches is a perfect match for our needs.

In **graph drawing**, the objective is to sketch a graph in the plane in a visually pleasing manner. There is an annual conference devoted to this topic, and numerous algorithms and applications can be found in the proceedings, eg. [12]. Possible objectives include non-overlapping vertices, few crossing edges and short edge lengths. The similarity data in our ordination problem can be interpreted as a weighted graph, in which objects are vertices and non-zero similarities are edges with similarity values as weights. Graph drawing techniques can now be applied to this graph.

However, the visual display of the graph of similarities is not of paramount concern. As will become clearer when we describe the visualization process in Section 5, the principle visual paradigm in VxInsight is the implicit clustering generated by the ordination process. In fact, the edges in the graph of similarity values might never be displayed. For this reason, graph drawing is not the right paradigm for our ordination process.

Another possible approach would be to use one of the many **clustering** algorithms (see, for example, the survey paper [27]). Specifically, large clusters could be identified and relegated to different portions of the geometric space. The subclusters within them could be identified and ordinated recursively.

Although appealing, this approach goes against the guiding philosophy of VxInsight. We feel that the human visual system is better than any algorithm at identifying patterns and trends. We don't want the clusters to be imposed by an ordination algorithm, but rather identified by the analyst as an implicit product of the ordination algorithm. To put it another way, an ordination algorithm that is based upon the detection of clusters may miss other features that are equally important to the analyst.

A third possible approach to the ordination problem is to use techniques from **multidimensional scaling** (MDS) (see, eg. [4]). The fundamental problem in MDS is to find a low-dimensional ordination for a set of objects that is consistent with some pairwise distance information.

Although this problem is closely related to our own, there are two reasons why we chose not to use techniques from this field for VxInsight. First, we expect our similarity data to be of low fidelity – mere hints about what is desired. Thus, we don't have any good input concerning distances to input to MDS. Second, the techniques used in MDS are computationally intensive, and we wish to be able to handle very large datasets, ideally in real time.

3.3. The Ordination algorithms in VxInsight

VxInsight currently contains two ordination algorithms with complimentary strengths and weaknesses. The first involves eigenvectors of a Laplacian matrix and is closely related to multidimensional scaling. The second is a force directed placement algorithm, similar to molecular dynamics, in which objects move about under attractive and repulsive forces. The eigenvector approach has the attractive property that it finds the global minimizer of a reasonable objective function. However, the ordinations it produces tend to be too tightly clustered, and so not ideal for interactive visualization. Force directed placement produces more visually appealing ordinations, but it tends to get stuck in local minima, and so does

not produce the best possible ordination. In our experience, the combination of these two algorithms is better than either alone.

3.3.1. Laplacian Eigenvectors One way to force similar objects to be close together is to minimize an appropriate penalty function. Many such functions are possible. However, many also lead to intractable computational problems. The particular approach described here reduces to a symmetric eigenvalue problem; a problem for which good software and algorithms are available.

Consider the following penalty function on the n objects

$$\text{Cost} = \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij} d_{ij}^2, \tag{1}$$

where s_{ij} is the non-negative similarity between objects i and j and d_{ij} is the geometric distance between them. Minimizing this function will encourage highly similar objects to be close together. However, merely minimizing Eq. 4 leads to a poorly phrased mathematical problem. Several constraints need to be added. First, note that translations of the full set of objects doesn't alter the cost. We can resolve this by adding constraints of the form $\sum_i x_i = 0$ and $\sum_i y_i = 0$, where x_i and y_i are the x and y coordinates for object i.

With these translational constraints, the minimum cost is trivially obtained by placing all the objects at the origin. This uninteresting solution can be avoided by adding the constraints $\sum_i x_i^2 = 1$ and $\sum_i y_i^2 = 1$. The solution to the resulting minimization problem will place all the objects along the main diagonal $x = y$. This can be avoided by adding a constraint of the form $\sum_i x_i y_i = 0$. Putting these constraints together, we have the following well posed minimization problem.

$$\begin{aligned}
&\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}\{(x_i - x_j)^2 + (y_i - y_j)^2\} \qquad (2)\\
&\text{Subject to :}\\
&\qquad \sum_{i=1}^{n} x_i = 0 \quad \text{and} \quad \sum_{i=1}^{n} y_i = 0\\
&\qquad \sum_{i=1}^{n} x_i^2 = 1 \quad \text{and} \quad \sum_{i=1}^{n} y_i^2 = 1\\
&\qquad \sum_{i=1}^{n} x_i y_i = 0
\end{aligned}$$

As discussed in [14], the solution to this problem involves the *Laplacian matrix* of the similarity function, L. Values in this matrix are defined as follows.

$$L(i,j) = \begin{cases} -s_{ij} & \text{if } i \neq j \\ \sum_{j=1}^{n} s_{ij} & \text{if } i = j \end{cases} \tag{3}$$

Denote the n eigenvectors of L by u_i, where the corresponding eigenvalues are ordered from smallest to largest. Then the solution to Eq. 5 can be shown to be $x = u_2$ and $y = u_3$ (see, for example, [14]).

The eigenvector problem has been well studied and several good software tools for this kind of calculation exist. We have chosen to use the ARPACK code by [19] due to its ability to limit the memory requirements. Using ARPACK with Tchebycheff polynomial preconditioning, we were able to ordinate a database with 2.4 million objects in less than a day on a high-end workstation.

Although Laplacian eigenvectors are computable in reasonable time and achieve the global minimizer of a plausible objective function, they have several shortcomings for our purposes. First, there are degenerate eigenvectors unless the graph with nonzero similarity values is *biconnected*. That is, the graph can't be divided into unconnected pieces by removing a single edge. In practice, this shortcoming can be handled in one of two ways. Either the biconnected components of the graph are handled up front (for a linear time algorithm, see, for example, [1]), or edges can be added to make the graph biconnected. Note that if a graph is connected, adding edges to shortcut all length-two paths will make it biconnected.

The second and more serious problem associated with Laplacian eigenvectors concerns the type of ordination they produce. In our experience, the objects tend to be highly clustered – so much so that navigation and interpretation are impaired. So for large datasets, we have found this approach to be unsuitable without some modification.

3.3.2. Force Directed Placement In the second ordination algorithm currently implemented as part of VxInsight, objects are moved about under the influence of attractive and repulsive forces. Specifically, each object is attracted towards the objects to which it is similar. The strength of this force is proportional to the similarity value. Simultaneously, a repulsive force exists between each pair of objects that are too close together. There are many possible variations of force laws within this basic model.

This approach, known as *force directed placement* has several attractive features, and has been advocated by several authors, eg. [9] and [13]. It generally produces ordinations that are attractive for interpretation and navigation. It is also good for incremental ordinations, where a few new objects are being added to an existing landscape.

However, this method also has limitations. Unlike the Laplacian eigenvector approach, force directed placement is sensitive to initial starting conditions and can generate a different answer with only small differences in initial data. Further, it can get trapped in a local minimum, and never find the globally optimal solution. Also, these methods can be very slow to compute, although tricks from the computational physics community can reduce computation time [2]. In particular, VxInsight uses a density grid technique to reduce the asymptotic running time from $O(n^2)$ (in a brute force approach) to $O(n)$.

Despite their individual drawbacks, together the Laplacian eigenvector algorithm and force directed placement complement each other. We have had success by first using eigenvectors and then refining the ordination using force simulations. In practice, force directed placement spreads the tightly clumped eigenvector ordination very nicely. Also, by initializing force directed placement with the globally good ordination from the eigenvector approach, the problem of multiple minima is reduced. Pairing these complementary methods produces an acceptable ordination, which requires only modest computing resources.

4. Geometric Placement of the Objects

A guiding motivation behind VxInsight is the use of geometric proximity in the graphical user interface to capture relationships between objects. Thus, a critical step is the mapping of objects to geometric coordinates in such a way that related documents are kept close together. The ability to extract useful information from the tool depends upon the quality of this *ordination*. The properties of a good ordination depend upon the nature of the data and the questions being asked, so it is our conviction that no single approach is a panacea. With domain-specific expertise, an analyst may be able to provide a better ordination than any general purpose tool. For this reason, we allow the analyst to choose among several ordination algorithms, or to bypass our algorithms and provide coordinates directly.

4.1. The Similarity Function

Input to the ordination algorithms in VxInsight consists of a *similarity* function $s : O \times O \rightarrow \mathbb{R}$, which maps object pairs to non-negative real numbers. Larger values imply that the two objects are more similar and hence should be located closer together. For large datasets, we expect most of the similarity values to be zero, so the data representing the similarity relationship will be sparse.

The details of the similarity function depend upon the application. Some simple examples of the data that could be used for similarity generation include the following.

1. Common keywords in documents.
2. Identical vocabulary within documents.
3. Citation links between scientific papers or patents.
4. Direct links in web documents.
5. Financial transaction links between corporations.
6. Membership in common organizations among individuals.

More generally, because any relational database includes information that couples objects, the different relational fields can be summed or combined in more complex ways to generate similarity values. Of course, more sophisticated methodologies, like latent semantic indexing, [7] could also be used. Obviously, the analyst should select a function that is appropriate for the kinds of questions being addressed.

4.2. Classes of Ordination Algorithms

Given a similarity function, the goal of the ordination process is to place the objects in a geometric space so that similar items are close together, and dissimilar objects are far apart. As we discussed above, for VxInsight we want a two-dimensional ordination, but the problem can be phrased in any dimension. Various approaches have been proposed for this and closely related problems; see, for example, the literature on self-organizing maps, eg.

[18]. Much of the prior work can be grouped into one of the three categories we discuss below. However, none of these general approaches is a perfect match for our needs.

In **graph drawing**, the objective is to sketch a graph in the plane in a visually pleasing manner. There is an annual conference devoted to this topic, and numerous algorithms and applications can be found in the proceedings, eg. [12]. Possible objectives include non-overlapping vertices, few crossing edges and short edge lengths. The similarity data in our ordination problem can be interpreted as a weighted graph, in which objects are vertices and non-zero similarities are edges with similarity values as weights. Graph drawing techniques can now be applied to this graph.

However, the visual display of the graph of similarities is not of paramount concern. As will become clearer when we describe the visualization process in Section 5, the principle visual paradigm in VxInsight is the implicit clustering generated by the ordination process. In fact, the edges in the graph of similarity values might never be displayed. For this reason, graph drawing is not the right paradigm for our ordination process.

Another possible approach would be to use one of the many **clustering** algorithms (see, for example, the survey paper [27]). Specifically, large clusters could be identified and relegated to different portions of the geometric space. The subclusters within them could be identified and ordinated recursively.

Although appealing, this approach goes against the guiding philosophy of VxInsight. We feel that the human visual system is better than any algorithm at identifying patterns and trends. We don't want the clusters to be imposed by an ordination algorithm, but rather identified by the analyst as an implicit product of the ordination algorithm. To put it another way, an ordination algorithm that is based upon the detection of clusters may miss other features that are equally important to the analyst.

A third possible approach to the ordination problem is to use techniques from **multidimensional scaling** (MDS) (see, eg. [4]). The fundamental problem in MDS is to find a low-dimensional ordination for a set of objects that is consistent with some pairwise distance information.

Although this problem is closely related to our own, there are two reasons why we chose not to use techniques from this field for VxInsight. First, we expect our similarity data to be of low fidelity – mere hints about what is desired. Thus, we don't have any good input concerning distances to input to MDS. Second, the techniques used in MDS are computationally intensive, and we wish to be able to handle very large datasets, ideally in real time.

4.3. The Ordination algorithms in VxInsight

VxInsight currently contains two ordination algorithms with complimentary strengths and weaknesses. The first involves eigenvectors of a Laplacian matrix and is closely related to multidimensional scaling. The second is a force directed placement algorithm, similar to molecular dynamics, in which objects move about under attractive and repulsive forces. The eigenvector approach has the attractive property that it finds the global minimizer of a reasonable objective function. However, the ordinations it produces tend to be too tightly clustered, and so not ideal for interactive visualization. Force directed placement produces more visually appealing ordinations, but it tends to get stuck in local minima, and so does

not produce the best possible ordination. In our experience, the combination of these two algorithms is better than either alone.

4.3.1. Laplacian Eigenvectors One way to force similar objects to be close together is to minimize an appropriate penalty function. Many such functions are possible. However, many also lead to intractable computational problems. The particular approach described here reduces to a symmetric eigenvalue problem; a problem for which good software and algorithms are available.

Consider the following penalty function on the n objects

$$\text{Cost} = \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij} d_{ij}^2, \tag{4}$$

where s_{ij} is the non-negative similarity between objects i and j and d_{ij} is the geometric distance between them. Minimizing this function will encourage highly similar objects to be close together. However, merely minimizing Eq. 4 leads to a poorly phrased mathematical problem. Several constraints need to be added. First, note that translations of the full set of objects doesn't alter the cost. We can resolve this by adding constraints of the form $\sum_i x_i = 0$ and $\sum_i y_i = 0$, where x_i and y_i are the x and y coordinates for object i.

With these translational constraints, the minimum cost is trivially obtained by placing all the objects at the origin. This uninteresting solution can be avoided by adding the constraints $\sum_i x_i^2 = 1$ and $\sum_i y_i^2 = 1$. The solution to the resulting minimization problem will place all the objects along the main diagonal $x = y$. This can be avoided by adding a constraint of the form $\sum_i x_i y_i = 0$. Putting these constraints together, we have the following well posed minimization problem.

$$\begin{aligned}
&\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}\{(x_i - x_j)^2 + (y_i - y_j)^2\} \qquad (5)\\
&\text{Subject to :}\\
&\qquad \sum_{i=1}^{n} x_i = 0 \quad \text{and} \quad \sum_{i=1}^{n} y_i = 0\\
&\qquad \sum_{i=1}^{n} x_i^2 = 1 \quad \text{and} \quad \sum_{i=1}^{n} y_i^2 = 1\\
&\qquad \sum_{i=1}^{n} x_i y_i = 0
\end{aligned}$$

As discussed in [14], the solution to this problem involves the *Laplacian matrix* of the similarity function, L. Values in this matrix are defined as follows.

$$L(i,j) = \begin{cases} -s_{ij} & \text{if } i \neq j \\ \sum_{j=1}^{n} s_{ij} & \text{if } i = j \end{cases} \tag{6}$$

Denote the n eigenvectors of L by u_i, where the corresponding eigenvalues are ordered from smallest to largest. Then the solution to Eq. 5 can be shown to be $x = u_2$ and $y = u_3$ (see, for example, [14]).

The eigenvector problem has been well studied and several good software tools for this kind of calculation exist. We have chosen to use the ARPACK code by [19] due to its ability to limit the memory requirements. Using ARPACK with Tchebycheff polynomial preconditioning, we were able to ordinate a database with 2.4 million objects in less than a day on a high-end workstation.

Although Laplacian eigenvectors are computable in reasonable time and achieve the global minimizer of a plausible objective function, they have several shortcomings for our purposes. First, there are degenerate eigenvectors unless the graph with nonzero similarity values is *biconnected*. That is, the graph can't be divided into unconnected pieces by removing a single edge. In practice, this shortcoming can be handled in one of two ways. Either the biconnected components of the graph are handled up front (for a linear time algorithm, see, for example, [1]), or edges can be added to make the graph biconnected. Note that if a graph is connected, adding edges to shortcut all length-two paths will make it biconnected.

The second and more serious problem associated with Laplacian eigenvectors concerns the type of ordination they produce. In our experience, the objects tend to be highly clustered – so much so that navigation and interpretation are impaired. So for large datasets, we have found this approach to be unsuitable without some modification.

4.3.2. Force Directed Placement In the second ordination algorithm currently implemented as part of VxInsight, objects are moved about under the influence of attractive and repulsive forces. Specifically, each object is attracted towards the objects to which it is similar. The strength of this force is proportional to the similarity value. Simultaneously, a repulsive force exists between each pair of objects that are too close together. There are many possible variations of force laws within this basic model.

This approach, known as *force directed placement* has several attractive features, and has been advocated by several authors, eg. [9] and [13]. It generally produces ordinations that are attractive for interpretation and navigation. It is also good for incremental ordinations, where a few new objects are being added to an existing landscape.

However, this method also has limitations. Unlike the Laplacian eigenvector approach, force directed placement is sensitive to initial starting conditions and can generate a different answer with only small differences in initial data. Further, it can get trapped in a local minimum, and never find the globally optimal solution. Also, these methods can be very slow to compute, although tricks from the computational physics community can reduce computation time [2]. In particular, VxInsight uses a density grid technique to reduce the asymptotic running time from $O(n^2)$ (in a brute force approach) to $O(n)$.

Despite their individual drawbacks, together the Laplacian eigenvector algorithm and force directed placement complement each other. We have had success by first using eigenvectors and then refining the ordination using force simulations. In practice, force directed placement spreads the tightly clumped eigenvector ordination very nicely. Also, by initializing force directed placement with the globally good ordination from the eigenvector approach, the problem of multiple minima is reduced. Pairing these complementary methods produces an acceptable ordination, which requires only modest computing resources.

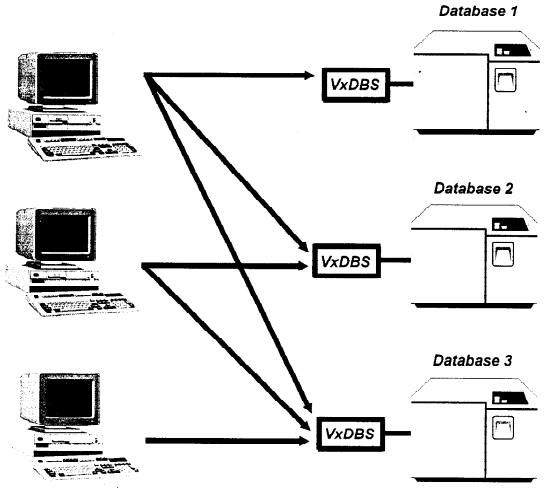

Figure 1. The VxInsight architecture allows users to connect to multiple databases and database servers to handle multiple clients

5. Software Architecture and Interface

In this section we will describe some of the physical architecture behind the system, including the client/server interface to the database engine, preparation of data, and the relationship of these processes to the user interface. A description of the interface functionality will explain our techniques for providing interaction at multiple levels of detail. We will also discuss how SQL is used to directly access the database for user specific queries, and how the results of these queries are graphically displayed. We will also describe VxInsight's various mechanisms to allow customization and to dynamically handle a wide variety of datatypes. In the last subsection a typical analysis session will be discussed. We include this example because we believe it demonstrates how VxInsight's true power comes from its synthesis of multi-disciplinary techniques.

5.1. VxInsight Physical Model

During the development of VxInsight, we relied heavily on layering strategies that allows different modules to be plugged in with minimal effect to the rest of the code. Knowing that this tool may be connected to a wide variety of databases, we designed lightweight database *monitors* that communicate with VxInsight through a standard protocol. Fig. 1 shows the connections between the databases and the clients using the visualization. In general, the analyst may connect to several databases and each database may have multiple

clients. The client/server connections are based on Tcl sockets which allow cross-platform communications.

Since VxInsight uses sockets, the database server can be widely separated from the visual client. This transparency of location can be used in several ways. Companies often have centralized databases that need to be tapped by analysts from their desktop workstations. Another alternative is for a data provider to sell access to their database and have customers from around the world connect to it using their VxInsight clients.

5.2. Data Preparation

Typically, analysts want to explore portions of what may be a very large amount of data. An analyst may have access to all patents granted between 1987-1997, but, for the moment, may be interested only in the progress and trends associated with computer storage devices. To cull the database down the analyst can use traditional methods already in her knowledge toolbox such as keywords, categories, references, etc. Now that the database contains a subset of perhaps 50,000 patents on storage devices and contributing technologies, VxInsight's work begins.

The initial step is to ordinate the data as discussed in Section 4. The result of this step is simply a flat file of unique IDs and (x, y) coordinates. The database administrator now loads this file into the database. To finish the data preparation, the analyst specifies the location of each data field by editing a configuration file that specifies the machine name, database table, and field name of each data item to be used. In this way, we allow the data to stay where it is, unlike other tools that force the user to move and reformat the data into the tool's native format. With VxInsight the analyst simply tells the tool where to find the data and then VxInsight will contact the database(s) and gather up the information. Once the gather operation is finished, the interaction with the tool begins.

5.3. User Environment

To aid in the exploration of data, VxInsight provides an interactive 3D environment as shown in Fig. 2. This environment allows the analyst full control of the viewing angle and distance from the data terrain. Control of the current view is based on the intuitive mouse actions listed below.

Mouse forward – Move closer to terrain
Mouse backward – Pull away from terrain
Mouse Left – Rotate terrain counter-clockwise
Mouse Right – Rotate terrain clockwise

Currently, these are the only degrees of freedom that the analyst has available. Internal user studies indicated that too many degrees of freedom only disoriented the analyst. In future versions, the advanced user may access additional viewing parameters through meta-keys.

The real power of information visualization comes with the capability to view the data on different scales. Once the big picture becomes clear, the ability to zoom in and explore the

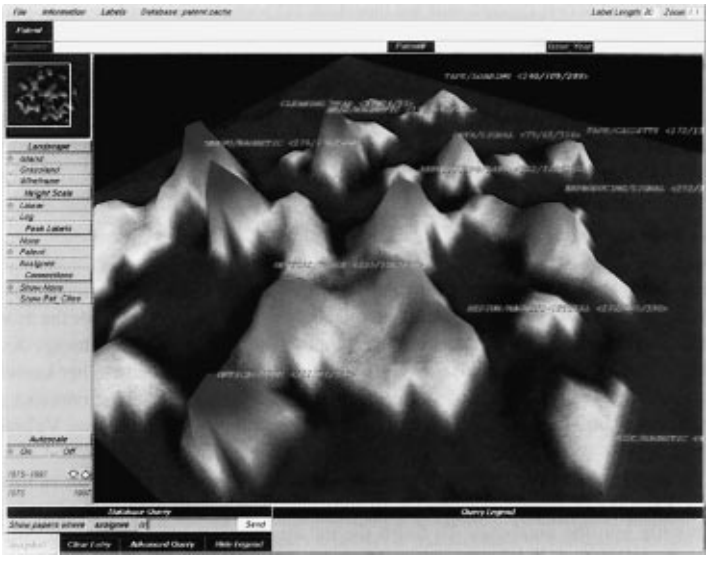

Figure 2: The mountain terrain metaphor provides an intuitive exploration environment.

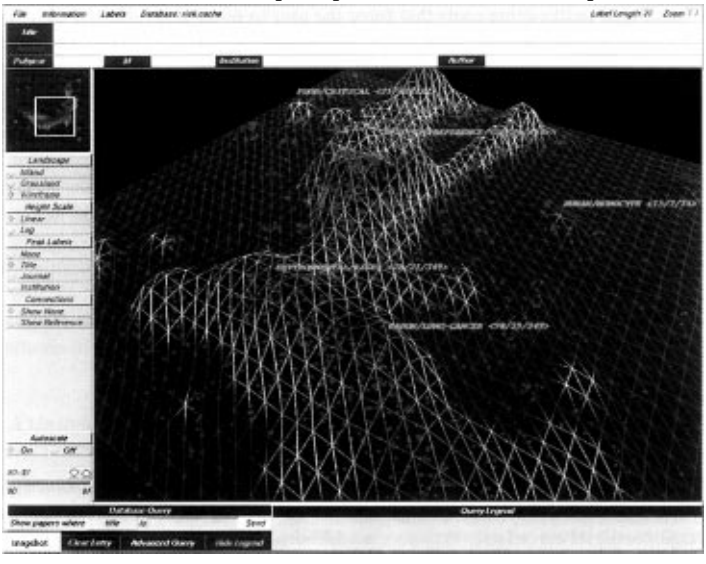

Figure 3: Dynamic terrain generation: Height is based on the density of data objects.

Figure 4: Mulit-resolution exploration with detail on demand.

contributing factors provides the analyst with a powerful deductive tool. Fig. 4 shows how

global level structure can be decomposed into mid-level connections, and then to micro-level details. The images displayed represent discrete levels of resolution, but when using the tool the viewing scale changes in a continuous fashion.

As with other controls, the zooming mechanism is simple and straightforward. The analyst picks an area of interest and clicks on it to zoom in. Holding the first button down provides a fluid transition to higher viewing resolutions. To zoom out, simply press the second mouse button. A typical analyst will change the viewing resolution quite often during a session of data exploration.

VxInsight provides three options for rendering mountain landscapes. The most popular is the transparent wireframe rendering, depicted in Fig. 3, which lets the analyst see the density of data elements below the mountains and the connection networks on higher levels. As shown in Fig. 2, the user can also view the terrain as solid mountains, which can displayed in two different color contrast schemes.

To explain the nature of the landscape, VxInsight dynamically generates labels for the most significant mountain peaks. It does this by looking for the most common words in a particular database field among the objects comprising a mountain. The analyst can dynamically change the field being used for labels. The peak label provides a rough estimate of the content of a mountain. The label displays the two most common words along with the number of times they occurred and the number of items sampled beneath the peak. The text processing required for peak labeling can be time consuming if a large number of objects are involved. To address this problem, VxInsight allows the user to specify a maximum number of objects to include in the processing. This results in only a sampling of objects being considered, trading off label quality for run-time. An example of the peak information strings can be seen in Fig. 2.

VxInsight can also display directional links between data objects. A dataset on ‘personnel interactions’ may have dynamic connection fields such as ‘Phone calls’ and ‘Lunches’. If the objects are scientific papers, the connections could be citation links. When the user selects one of the connections for display, directional links between objects will appear as can be seen in Fig. 4. These links give the analyst information about the structure of the data. On the highest level there may be so many connections that not much information can be extracted, so this option is mainly used when the current resolution contains a manageable number of objects.

Emerging trends, technological dependencies, and information propagation can all be detected using VxInsight with temporal data. The analyst can limit the display to a restricted time period to visualize the temporal evolution of the data. The sequence of images in Fig. 5, shows some of this feature’s power. By sliding the visualization through time, interesting aspects of the data emerge. In this case we see a pronounced technological shift in laser research. (This example will be discussed in more detail below.) By querying the database we can also identify who is developing those technologies. Fig. 5 also demonstrates the effect of information propagation. The research starts in Japan (blue), comes the US National Labs (orange,green) and then propagates to Germany (yellow).

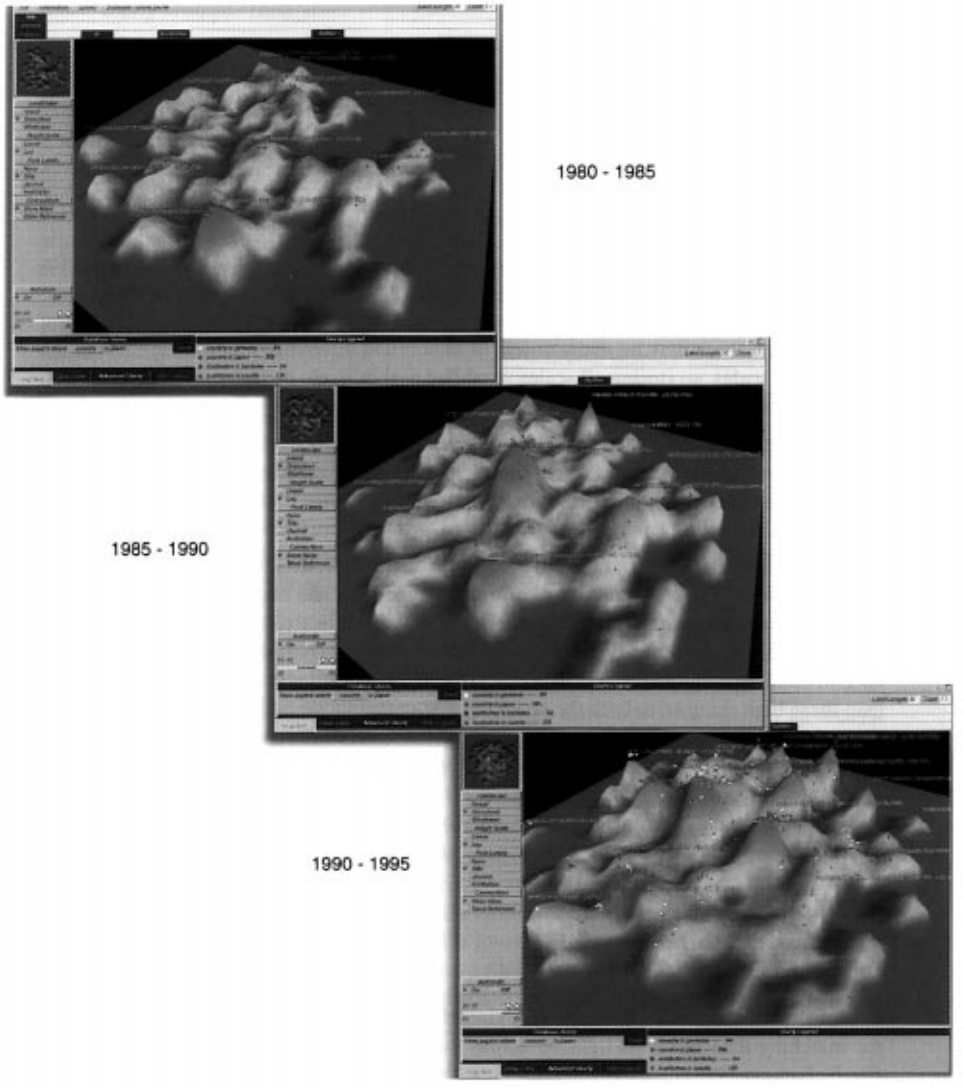

Figure 5: Using database queries and temporal sequencing in combination.

5.4. *Multi-resolution Viewing*

One of the guiding principles behind VxInsight was the need to provide multiple resolutions of the data set. Providing seamless traversal through viewing resolutions allows greater comprehension of the data on the whole. The principle drawback to the multi-resolution concept is its computational cost. Allowing the user to have all these different views involves either dynamically computing the views on the fly or pre-processing the data and storing various resolutions to be called up at run-time[1].

With visualization of large datasets in mind, VxInsight has chosen the dynamic route for computing multi-resolution views. When exploring large information spaces, much of the memory is allocated to holding the data[2], leaving little for the visualization. Also, depending on the dataset, there may be many orders of magnitude difference from the highest to the lowest view scales, so storing all scales would be prohibitively memory intensive. Instead, VxInsight computes and provides *detail on demand* [23].

Detail on demand refers to the postponing of detail computation until it is requested by the user. First, an overview of the entire collection is presented. Given an overview, the analyst can now zoom in on areas of interest. Only these areas are recomputed at higher resolution.

Software tools must be available to a large customer base. In the development of VxInsight steps were taken to minimize platform dependencies and exclusions. The VxInsight environment may appear to require extensive graphics hardware, but in fact uses polygons quite sparsely. To dynamically generate the terrain at various resolutions we overlay a regular grid on the currently viewable data objects. All objects fall into one of the grid bins, and then these bins are used in both the terrain height computation and labeling algorithm. By using this approach we find a good compromise between modest resolution and interactive rates for dynamic terrain generation. The regular grid technique requires only a constant number of polygons no matter what the resolution level. To be precise, the terrain consists of 3200 triangles, all t-striped to improve performance. This means that a modest PC (266 Mhz) with an OpenGL graphics card is quite capable of running VxInsight.

5.5. *Querying the Database*

One of the most powerful features of VxInsight is the ability to query a database and see the matching data elements light up across the terrain with their respective colors, as in Fig. 5. For instance, at the highest level view, the analyst can use the SQL interface to make a database query about all the papers that were published in Japan. These might show up as blue dots lying on the terrain directly above the associated papers. A second and third query might ask to mark the papers from Germany and from the United States.

Now one can quickly pin-point the concentrations of these countries in the research area being explored. When used with the time slider, visual specifications of matching queries can be used to track the ebb and flow of competitive advantage as represented in the published record.

5.6. Using Multiple Databases

Of course, most analysts will work with more than one database. They may choose to work for an extended time with one database, then turn to another one for equally long periods of work. However, it is more likely that they will want to switch between correlated databases during the same session. VxInsight provides features to support this switching, invisibly to the user.

When a dataset is accessed, a simple descriptor file is consulted. This descriptor file allows the analyst to customize the menus for a particular research focus. For example, when a dataset of web pages is being viewed, the informational display may have text widgets labeled 'Title' and 'Web Address'. The peak labels can be set to either 'Title' or 'Company' and the 'Connections' menu allows us to show 'Web Links'. Now by simply opening a dataset based on financial dealings, the entire informational display and menuing options transform. The information display is now composed of text widgets reading 'Transaction Type', 'Amount', and 'Date'. The peak label options are now 'Buyer', 'Seller','Institution', and the 'Connections' menu allows us to show either 'Country' or 'Bank' links in the data. The SQL query options will also be automatically updated. The SQL query window allows boolean conjunctions and negations using the table names of the current database. These are immediately presented to the analyst, which is very useful when initially learning to use a new database, or when you come back to a database after an extended period of time and have forgotten the field names.

5.7. An Example Session

Now that some of the features in VxInsight have been introduced, let's follow an analyst using the system. This example concerns a semiconductor laser technology called Vertical Cavity Surface Emitting Lasers (VCSELs). The pertinent papers would first have been extracted from a large database of scientific documents, and processed as described earlier. Similarities were constructed using co-citation analysis [21].

The analyst starts the local VxInsight client software, which automatically connects to the VxInsight database server, which is already connected to the database. The VCSEL database is requested, and a highest-level view of that terrain is presented. The analyst asks to have the peaks labeled using words from the journal names. These clues begin to explain the visible landscape. Various concepts have their own peaks, for example fabrication technologies, applications of VCSELs, and some technical titles indicating field jargon. Now the analyst specifies that the peaks be labeled with the institution names and see many labels like 'AT&T', 'Sandia', and 'Marietta'.

With this information, our analyst makes a query asking the system to mark all the papers with an author from Sandia National Labs. Suddenly, it becomes clear that fabrication technologies are important to Sandia. Now the analyst zooms down into that region to reach the individual papers. She is particularly interested in finding a seminal paper, maybe one that started the whole field. Such a paper will be referenced by most of the important papers under the mountain. The analyst requests that the system draw lines to show the linkages between the citing and the cited papers. Sure enough, a few papers stand out as much more important than others.

Our analyst wonders if most of the work is being done in the US, so she zooms back out to a higher level and makes an SQL query asking for papers from US institutions. Many of the papers are marked, but it is now obvious that the US does not have a monopoly on this technology. Now curious, the analyst asks about Japanese papers, and then perhaps other countries, until she thinks to ask if the Germans have worked in this field. Sure enough, they have a strong presence.

Ever curious, our analyst wonders if research into this area has always been spread around the world, or if the leadership has changed through time. To test this idea, she selects a time period early in the history of the technology. The early papers are almost all from Japan. Moving forward in time she notes a strong surge of American interest, though the Japanese effort never really waned. Moving to the most recent papers, she notices an almost instantaneous burst of German publications. She is now quite interested in this burst of German activity so she now investigates the German papers to see what institutions they come from, what papers they reference, and who references them. At this point, she may stop using VxInsight and choose to switch to some of the more conventional tools at her disposal to investigate this phenomenon. The point we want to make here is that she uncovered a trend by simply exploring the data. Her more conventional tools are only useful when she already knows what she's looking for.

An actual analysis session in the field of VCSELs, though not identical to this little story, did happen and was generally in the spirit as presented. Since that time, several other studies in various fields have been undertaken, as will be mentioned in the next section.

6. Applications

The development of VxInsight was motivated by a simple, and universal, question: Where do we put the next research dollar for the most impact? In thinking about an approach to this question, it became clear that an underlying prerequisite must be the development of a more robust understanding of how we got to where we are today. While no scientist, administrator, or policy analyst can completely *know* the world of science, it is necessary to understand the technical details sufficiently to follow the evolution of the scientific disciplines to direct research funding intelligently. An investment strategy must weigh many questions to evaluate proposals. For example,

- What prior work was necessary before a breakthrough became possible?
- How do fields converge or diverge over time?
- How do new insights propagate across disciplines?
- Who are the central figures in an evolving research thread?
- In any given area, what are our competitors doing (where competitors can be individuals, institutions, or even countries)?

To begin to understand such structural issues, an obvious place to look is in the record of scientific progress documented by research publications. The evolution of ideas is contained in these papers, sometimes fully disclosed by the citation links, but sometimes

only implicitly acknowledged through indirect chains of citations. Citation analysis is a rich source of information about scientific researchers and disciplines [21].

If the body of a paper contains its contributions, the paper's citations provide the recorded context – the environment in which the research evolved – and are the author's acknowledgment of the intellectual heritage leading to the contribution presented in the paper itself. VxInsight was first conceived as a tool for examining and exploring the structure in those citation linkages. Our objective was to provide a tool with which an analyst could examine a very large set of citation information, follow the evolution of individual technologies, and, potentially, gain insight into where those technologies might be heading.

One example of the application of VxInsight to the investigation of scientific disciplines was outlined in Section 5.7. The tool has been used for several such studies. For the VCSELs study sketched above, we began with a subset of relevant papers from the Science Citation Index from the Institute for Scientific Information. VxInsight allowed an analyst to discover who the original developers of the technology were, who then took the technology and made significant strides to improve it, and at what level other organizations and countries have continued research hoping to eventually build the world's most efficient low-power light source. Most revealing, she uncovered a systematic shift of research leadership around the world as the technology matured. This exploration process required no specialized knowledge about the field.

Usually the world-wide flow of technology is good, but sometimes we would rather our secrets stayed at home. For example, VxInsight was used at Sandia to study nuclear proliferation [17]. As with the VCSELs study, scientific papers in nuclear technology and their citation links were used to create an initial display of the structure of the nuclear technology literature. Then, text analysis tools were used to determine similarities between the scientific papers and other public sources discussing nuclear technologies, such as international news stories. With these new associations between the underlying science and the popular reports, those news stories were added to the landscape of scientific specialties. This coupling revealed which stories were related to sensitive information and provided a powerful, new intelligence tool. This example illustrates the flexibility of VxInsight for various types of similarity analysis.

Another group at Sandia is using VxInsight to identify potential strategic partnerships that can benefit our laboratory. They compare the relationships between external publications from other organizations with the published research of Sandia scientists. The intersection of these records, presented in a VxInsight landscape, reveals potential partners. Often the individual scientists are already aware of researchers at the indicated institutions. However, without VxInsight, managers and technology transfer experts would be unlikely to find these connections on their own. This is a powerful attribute of VxInsight – non-experts can quickly gather information that only experts, in their narrow fields, knew about or had internalized. Broader strategic visions are enabled as this information flows to managers, funding agents, and inter-company negotiators.

Scientific papers are not the only objects that have natural linkage structures. Linkages are ubiquitous in business, in government, in social organizations; indeed most human endeavors have aspects that can be represented with this abstraction. VxInsight allows a user to *see* these implicit structures and to begin to interpret them. What do the structures mean and are they relatively stable or just transient phenomena? Can we analyze a structure

to discover principles governing its creation and the couplings among the elements? That is, can we use these structural hints to create new and useful knowledge with practical application in decision making?

One obvious example is to apply VxInsight to patent citations, which are very similar to citations between scientific papers. With VxInsight, a patent analyst can quickly investigate questions like: where are competitors placing their efforts, and how does their intellectual property intersect with our own? Are there areas that offer potential for patent circling? Who is citing our patents, and what types of things have they developed? Are there emerging competitors or collaborators working in areas of importance to us?

Another example, which is currently being pursued by colleagues at Sandia, is multi-dimensional transactional analysis (for example, the record of electronic funds transfer). These databases have many fields, or tables, any combination of which might be used to study the structures within the record of transactions. Different similarity functions can be developed, each of which weights the relative significance of particular fields as needed for individual investigations. While different similarity functions allow for different views of the data, the entire database always remains accessible via SQL queries.

Government agencies and private businesses are both interested in understanding transactions, particularly financial transactions. The ability to quickly understand and follow ideas and leads through a database of transactions can have huge financial implications. Currency trading, planning for international development, and detecting world-wide trading patterns all require timely information, which can be presented and studied using VxInsight.

A number of additional applications of VxInsight have been considered. They include detecting medicaid fraud, product marketing research, counter-terrorism intelligence, monitoring the growth of national imports and exports, structuring the results of World-Wide Web searches. The broadest applications will surely involve abstract data, representations of concepts uncoupled from any physical manifestation. With this broader point of view, we tried to make sure that VxInsight could be used with very general classes of networks and databases.

Anywhere there is a relationship between data elements or anywhere an abstract relationship can be defined, VxInsight can be used to explore the relationships using the natural, intuitive landscape metaphor to organize and present the implicit structures. We feel that the flexibility of our visualization paradigm and the associated ordination software will allow for a large and diverse set of applications, many of which we can not yet envision.

7. Conclusions

VxInsight is a powerful and flexible tool for interacting with large collections of abstract information. It is particularly well suited for identifying structure and patterns in large datasets, a task that is difficult for traditional knowledge management tools.

VxInsight employs an intuitive landscape metaphor, which simplifies interpretation and navigation. Geometric proximity is used to convey similarity between objects, but the precise meaning of ‘similarity’ is under the control of the analyst. The tool allows for zooming to explore interesting regions in greater detail, which can reveal structure on multiple scales. An SQL connection to the database allows the retrieval of detailed information about the objects. Properties of the data can also be encoded into the landscape, allowing visualizations

of the distribution of objects matching a query, or the comparison of multiple queries. By design, no matter how many objects are in the database the geometry of the basic landscape display is simple, and so can be rendered quickly. More complex views involving large numbers of highlighted objects or many linkages between objects are obviously more time consuming to render. The calculation of suitable peak labels can also be expensive. But all this functionality runs at interactive frame rates on a modest PC for datasets with many tens of thousands of objects. We are working towards the ability to interact with millions of objects, but further advances in both algorithms and hardware will be required.

The software was designed to be easily reconfigurable to work with new and different types of data. It has been used at Sandia National Laboratories in patent analysis, nuclear nonproliferation, strategic planning, World Wide Web analysis, research prioritization and other applications. In all of these settings, VxInsight has allowed the analyst to overview large collections of data, to identify interesting patterns and trends, to ask unanticipated questions, and to explore possible answers.

We believe that VxInsight is a prototype of knowledge management tools, which will heavily rely on advanced visualization techniques. The human visual system is extraordinarily adept at surveying large sets of information and identifying trends and exceptions. The challenge is to present abstract information in a form that allows our visual talents to excel. Based upon our experience with VxInsight, we advocate the following principles for this task.

- Use a familiar metaphor.
- Enable navigation, but keep it simple.
- Provide mechanisms for information retrieval.
- Present information at different levels of detail.
- Include multiple, simultaneous display options.
- Make it run fast on large datasets.

8. Conclusions

VxInsight is a powerful and flexible tool for interacting with large collections of abstract information. It is particularly well suited for identifying structure and patterns in large datasets, a task that is difficult for traditional knowledge management tools.

VxInsight employs an intuitive landscape metaphor, which simplifies interpretation and navigation. Geometric proximity is used to convey similarity between objects, but the precise meaning of ‘similarity’ is under the control of the analyst. The tool allows for zooming to explore interesting regions in greater detail, which can reveal structure on multiple scales. An SQL connection to the database allows the retrieval of detailed information about the objects. Properties of the data can also be encoded into the landscape, allowing visualizations of the distribution of objects matching a query, or the comparison of multiple queries. By design, no matter how many objects are in the database the geometry of the basic landscape display is simple, and so can be rendered quickly. More complex views involving large

numbers of highlighted objects or many linkages between objects are obviously more time consuming to render. The calculation of suitable peak labels can also be expensive. But all this functionality runs at interactive frame rates on a modest PC for datasets with many tens of thousands of objects. We are working towards the ability to interact with millions of objects, but further advances in both algorithms and hardware will be required.

The software was designed to be easily reconfigurable to work with new and different types of data. It has been used at Sandia National Laboratories in patent analysis, nuclear nonproliferation, strategic planning, World Wide Web analysis, research prioritization and other applications. In all of these settings, VxInsight has allowed the analyst to overview large collections of data, to identify interesting patterns and trends, to ask unanticipated questions, and to explore possible answers. We believe that VxInsight is a prototype of knowledge management tools, which will heavily rely on advanced visualization techniques. The human visual system is extraordinarily adept at surveying large sets of information and identifying trends and exceptions. The challenge is to present abstract information in a form that allows our visual talents to excel. Based upon our experience with VxInsight, we advocate the following principles for this task.

- Use a familiar metaphor.
- Enable navigation, but keep it simple.
- Provide mechanisms for information retrieval.
- Present information at different levels of detail.
- Include multiple, simultaneous display options.
- Make it run fast on large datasets.

Acknowledgments

This work was performed at Sandia National Labs which is operated for the US DOE by Lockheed Martin Corp. under contract DE-AC04-94AL85000, and was supported by Sandia's LDRD program.

We are deeply indebted to Henry Small and David Pendlebury for helping define the scope of the work. We also benefited greatly from the input provided by Nancy Irwin, Helen Koller, yce Van Berkel, Kevi Boyack and Ken McCurley.

Notes

1. There exist hybrid schemes, notably wavelets, that precompute quadtrees that store averages and differences in the resolution levels. This allows memory consumption to be minimized at the cost of a small amount of computation at run-time. See, for example, [24].
2. Even when most of the data resides in the database, a subset is stored locally to generate peak labels and connection information at interactive rates.

References

1. Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1974). *The Design and Analysis of Computer Algorithms*. Reading, PA: Addison-Wesley.
2. Allen, M.P. and Tildesley, D.J. (1987). *Computer simulation of liquids*. Oxford: Oxford Science Publications.
3. Bar, F. and Borrus, M. (1993). The future of networking. Technical report, University of California, BRIE Working Paper, Berkeley, CA.
4. Borg, I. and Groenen, P. (1997). *Multidimensional Scaling: Theory and Applications*. Springer Series in Statistics.
5. Chuah, M.C., Roth, S.F., Mattis, J. and Kolojejchick, J. (1995). SDM: Selective dynamic manipulation of visualizations. *Proc. ACM Symp. User Interface Software and Technology*, (pp. 61–70). Pittsburgh, PA: ACM.
6. Cook, D. and Buja, A. (1997). Manual controls for high-dimensional data projections. *J. Computational and Graphical Statistics*, 6.
7. Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, R. (1990). Indexing by latent semantic analysis. *J. Amer. Soc. Information Science*, 41, 391–407.
8. Edelstein, H. (1997). Mining for gold. *Information Week*. (pp. 52–70).
9. Fruchtermann, T. and Rheingold, E. (1990). Graph drawing by force-directed placement. Technical Report UIUCDCS-R-90-1609, Dept. Computer Science, Univ. Illinois, Urbana-Champagne.
10. Ginchereau, B., Dunn, J. and Mitchell, L. (1997). Knowledge management solutions. *Info World*. (pp. 116–126).
11. Girardin, L. (1996). Mapping the virtual geography of the world-wide web. *Proc. Fifth International World Wide Web Conf.* Elsevier.
12. Goos, G., Hartmanis, J. and van Leeuwen, J. (Eds). (1997). *Proceedings of Graph Drawing '97*. Springer-Verlag.
13. Hendly, R.J., Drew, N.S., Wood, A.M. and Beale, R. (1995). Case study: Narcissus: Visualizing information. *Proc. InfoVis 95* (pp. 90–96). IEEE Computer Society Press.
14. Hendrickson, B. and Leland, R. (1995). An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J. Sci. Comput.*, 16, 452–469.
15. Honkela, T., Kaski, S., Kohonen, T. and Lagus, K. (1998). Self-organizing maps of very large document collections: Justification for the WEBSOM method. In I. Balderjahn, R. Mathar, and M. Schader (Eds.), *Classification, Data Analysis, and Data Highways*. Berlin: Springer. See http://websom.hut.fi/websom/.
16. Honkela, T., Kaski, S., Lagus, K. and Kohonen, T. (1996). Newsgroup exploration with WEBSOM method and browsing interface. Technical Report A32, Helsinki Univ. Tech. Laboratory of Computer and Information Science. See http://websom.hut.fi/websom/.
17. Irwin, N.H., van Berkel, J., Johnson, D.K. and Wylie, B.N. (1997). Navigating nuclear science: Enhancing analysis through visualization. Technical Report SAND97–2218, Sandia National Labs, Albuquerque, NM,.
18. Kohonen, T. (1997). *Self-Organizing Maps, Second Extended Edition*. Berlin: Springer.
19. Lehoucq, A.B., Sorensen, D.C. and Yang, C. (1998). *ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. Phildelphia, PA: SIAM.
20. Lin, X. (1997). Map displays for information retrieval. *J. American Soc. Information Sci.*, 48, 40–54.
21. Osareh, F. (1996). Bibliometrics, citation analysis and co-citation analysis: A review of literature ii. *LIBRI*, 46, 217–225.
22. Rathjens, D., Galgani, D., Kleinfeld, C. and Cary, C. (1996). MineSet user's guide. Technical Report 007–3214–002, Silicon Graphics, Inc., Mountain View, CA.
23. Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *Proc. IEEE Symp. Visual Languages '96*. IEEE.
24. Stollnitz, E.J., Derose, T.D. and Salesin, D.H. (1996). *Wavelets for Computer Graphics*. Morgan Kaufmann.
25. Tufte, E.R. (1992). *The visual display of quantitative information*. Graphics Press.
26. Visual insights. See http://www.visualinsights.com/.
27. Willett, P. (1988). Recent trends in hierarchic document clustering: a critical review. *Information Processing & Management*, 24, 577–597.
28. Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M., Schur, A. and Crow, V. (1995). Visualizing the non-visual: Spatial analysis and interaction with information from text documents. *Proc. 1995 IEEE Symp. Information Visualization* (pp. 51–58). IEEE. See http://www.thememedia.com/.